



AUTORES: Alberto Sánchez, Ana Belén García y Ángel Gordón

CICLO FORMATIVO DE GRADO SUPERIOR DE DESARROLLO DE APLICACIONES WEB

ÍNDICE

1. Abstract.....	3
2. Justificación del proyecto.....	4
3. Introducción.....	6
4. Plan de empresa.....	6
4.1 Análisis DAFO.....	7
5. Diseño web.....	9
5.1 Briefing.....	9
5.2 Branding.....	10
5.3 Mockups.....	11
6. Preparación del entorno de desarrollo.....	14
7. Desarrollo web.....	16
7.1 Estructura de la web.....	16
7.2 Funcionalidad.....	16
7.2.1 Index.....	16
7.2.2 Registro y Login.....	17
7.2.3 Tienda y productos.....	18
- Vista Individual de productos.....	18
- Vista general de productos.....	19
- Buscador.....	20
7.2.4 Sistema de carrito y compra.....	20
- Vista de carrito.....	21
- Proceso de compra y asignación de keys.....	21
7.2.5 Sistema de carrito y compra.....	22
- Mis datos.....	22
- Métodos de pago y dirección de facturación.....	23
- Formularios de pago y dirección de facturación.....	23
7.2.6 Panel de control.....	23
- Wishlist.....	23
7.3 Casos de uso.....	24
7.4 Portabilidad.....	33
7.5 Rendimiento.....	34
8. Diagrama de Gantt.....	36
9. Conclusión del proyecto.....	38
10. Bibliografía.....	39

1. Abstract

Hoy en día el mercado de los videojuegos está en auge, formando parte de una de las industrias con las mayores ganancias en el mundo (108.500 millones de dólares en 2021). Por esta razón se sigue invirtiendo grandes cantidades tanto en el desarrollo de los juegos por parte de las empresas como en la compra de los mismos por parte de los usuarios. Con la aparición de los portales de videojuegos se abrió un mundo de compraventa aún por explorar del cual queremos formar parte con "Game Nexus".

Esto lo queremos lograr con una página web de compra de claves, desarrollada con Symfony y Javascript entre otros lenguajes, así como logrando un despliegue de la propia página a la red para que cualquier persona del mundo pueda adquirir su producto.

Today, the video game market is booming, being part of one of the industries with the highest profits in the world (\$108.5 billion in 2021). For this reason, large amounts are still being invested both in the development of games by companies and in their purchase by users. With the emergence of gaming portals, a world of buying and selling has opened up that is still largely unexplored, and we want to be part of it with "Game Nexus."

We aim to achieve this with a website for purchasing game keys, developed using Symfony and JavaScript among other languages, as well as deploying the website to the internet so that anyone in the world can purchase their product.

2. Justificación del proyecto

El proyecto de Game Nexus ha sido desarrollado con la intención de plantear el desarrollo de una **web funcional y profesional** en el que se plantea el **empleo de diversas tecnologías adicionales** a las ya vistas durante el grado junto con un desarrollo sólido y profundo tanto de backend como de frontend. Para llevar a cabo este desarrollo, se ha decidido utilizar el **framework Symfony**, una de las herramientas más robustas y populares en el ámbito del desarrollo web.

Symfony es un **framework PHP** de alta eficiencia que permite construir aplicaciones web complejas de manera estructurada y mantenible. Fue creado por SensioLabs y ha ganado una gran popularidad debido a su **flexibilidad, escalabilidad y su enfoque en las mejores prácticas** del desarrollo de software. Entre sus características más destacadas se incluyen:

1. **Modularidad:** Symfony se basa en componentes reutilizables que pueden ser integrados de manera independiente o conjunta, facilitando la creación de aplicaciones personalizadas y optimizadas para necesidades específicas.
2. **Flexibilidad:** Permite a los desarrolladores configurar y personalizar su estructura según los requisitos del proyecto, sin imponer restricciones rígidas, lo que facilita la adaptación a diferentes tipos de proyectos.
3. **Escalabilidad:** Es ideal para proyectos que requieren crecimiento continuo, ya que su arquitectura está diseñada para manejar grandes volúmenes de datos y usuarios sin comprometer el rendimiento.
4. **Seguridad:** Symfony incluye múltiples mecanismos de seguridad, como la gestión de autenticación y autorización, protección contra ataques CSRF (Cross-Site Request Forgery), y cifrado de datos, asegurando que la aplicación sea resistente a vulnerabilidades comunes.
5. **Ecosistema Rico:** La amplia comunidad de desarrolladores y la abundancia de bundles y componentes preconstruidos permiten una integración más rápida de funcionalidades complejas, ahorrando tiempo y esfuerzo en el desarrollo.

Para manejar la capa de presentación en Game Nexus, hemos utilizado **Twig**, el **motor de plantillas nativo de Symfony**. Twig facilita la separación de la lógica de presentación de la lógica de negocio, permitiendo que se trabaje de manera más eficiente y organizada. Sus principales beneficios incluyen:

- **Simplicidad y Flexibilidad:** Twig utiliza una sintaxis sencilla y fácil de aprender que permite a los desarrolladores crear plantillas de manera rápida y eficaz.
- **Rendimiento:** Twig compila las plantillas a PHP puro, lo que hace que la ejecución sea extremadamente rápida y eficiente.
- **Seguridad:** Twig incluye mecanismos integrados para prevenir ataques de inyección de código y otros tipos de vulnerabilidades, asegurando que las aplicaciones sean más seguras.

En cuanto a la **gestión de datos**, Symfony integra **Doctrine**, un ORM (Object-Relational Mapper) que facilita la **interacción con la base de datos**. Doctrine permite trabajar con **datos como si fueran objetos PHP**, evitando la necesidad de escribir consultas SQL manualmente. Esto se traduce en un código más limpio y mantenible:

- **Mapeo Objeto-Relacional:** Doctrine mapea las tablas de la base de datos a clases de PHP, facilitando la manipulación de datos de manera orientada a objetos.
- **Gestión de Entidades:** Las entidades en Doctrine representan las tablas de la base de datos y permiten interactuar con los datos de forma estructurada.
- **Consultas:** Doctrine proporciona DQL (Doctrine Query Language), un lenguaje de consulta que opera sobre objetos y clases PHP, similar a SQL pero más orientado a objetos.

Al utilizar Symfony para el proyecto Game Nexus, se busca no solo construir una plataforma eficiente y segura, sino también aprovechar la oportunidad para **profundizar en el uso de tecnologías avanzadas y prácticas** de desarrollo profesional. Este enfoque asegura que el proyecto no solo cumpla con los requisitos funcionales y técnicos establecidos, sino que también se alinee con estándares de calidad reconocidos en la industria, promoviendo una base sólida para **futuros desarrollos y escalabilidad**.

En resumen, la elección de Symfony como framework principal para el desarrollo de Game Nexus subraya nuestro compromiso con la excelencia técnica y la adopción de tecnologías modernas, permitiendo un **desarrollo robusto y eficiente** que garantice tanto la **funcionalidad como la mantenibilidad** del proyecto a largo plazo.

3. Introducción

En el vasto universo de la programación, siempre **buscamos desafíos** que nos lleven más allá de nuestros límites. ¿Qué mejor manera de hacerlo que fusionando nuestra pasión por la tecnología con el emocionante mundo de los videojuegos?

Game Nexus no es solo una tienda online, es el resultado de un sueño hecho realidad. Desde nuestros primeros pasos en el desarrollo, nos propusimos crear algo más que un simple proyecto: queríamos construir una **plataforma que reflejará la profesionalidad y la pasión** que sentimos por lo que hacemos.

Así nació la idea de Game Nexus. Imagina una tienda virtual donde puedes explorar una **amplia gama de juegos** para todas las plataformas: PC, consolas y más. Sin embargo, no nos detenemos ahí. Queríamos ofrecer una experiencia completa, desde la **selección del juego perfecto hasta la comunidad activa** que lo respalda.

Con tecnologías de vanguardia en el backend y un frontend meticulosamente diseñado, hemos creado un espacio donde los aficionados a los videojuegos pueden encontrar todo lo que necesitan. **Game Nexus es el proyecto** que siempre hemos anhelado llevar a cabo, y ahora podemos ver como ha tomado forma con éxito.

4. Plan de empresa

Game Nexus es una tienda de videojuegos diseñada para ser un punto de encuentro para los amantes de los videojuegos, combinando la emoción de los juegos con una experiencia de usuario envolvente. El **objetivo principal** del proyecto es **establecer una identidad de marca sólida** que refleje nuestra pasión por la cultura del juego y nuestro compromiso con la comunidad de jugadores. Nuestra plataforma ofrecerá una amplia gama de videojuegos, noticias actualizadas y opiniones de la comunidad, todo en un solo lugar.

Nuestra **audiencia** objetivo incluye jugadores de todas las edades y géneros, comunidades de jugadores locales y en línea, entusiastas de la cultura geek y posibles colaboradores y socios de la industria de los videojuegos. Nos proponemos atraer y conectar con estos grupos ofreciendo una marca que se perciba como un verdadero **punto de encuentro** para los amantes de los juegos, donde la emoción y la creatividad se fusionan.

La **estrategia de marketing** de Game Nexus se centra en el desarrollo de una **identidad visual vibrante y moderna** que resuene con la comunidad de jugadores. Utilizaremos plataformas de redes sociales para interactuar con nuestra audiencia y promocionar productos. Además, implementaremos **estrategias de optimización de motores de búsqueda** (SEO) y marketing en motores de búsqueda (SEM) para aumentar nuestra visibilidad en línea. También organizaremos eventos comunitarios y colaboraremos con influencers y marcas dentro de la industria de los videojuegos para ampliar nuestro alcance y reforzar nuestra presencia en el mercado.

Para llevar a cabo este proyecto, necesitamos recursos tecnológicos que incluyen el desarrollo y mantenimiento de la plataforma web, sistemas de gestión de contenido y bases de datos. Nuestro **equipo** estará compuesto por desarrolladores web, diseñadores gráficos, especialistas en marketing y personal de atención al cliente. La **inversión inicial** será necesaria para el **desarrollo** del sitio web, así como para las actividades de **marketing** y los **eventos de lanzamiento**.

4.1 Análisis DAFO

Game Nexus presenta varias **fortalezas**. Una de ellas es su **identidad de marca distintiva**, que combina elementos modernos y nostálgicos, lo que la hace atractiva para una amplia gama de jugadores. Además, nuestro **enfoque comunitario** está diseñado para fomentar la participación activa de los usuarios y crear una comunidad vibrante. Ofrecemos una variedad de contenido, no solo en la venta de videojuegos, sino también en noticias, lo que enriquece la experiencia del usuario. Nuestro uso de **tecnologías modernas** nos permite ofrecer una experiencia de usuario intuitiva y envolvente.

Sin embargo, también enfrentamos algunas **debilidades**. Nuestra dependencia de la tecnología significa que debemos **mantener y actualizar** constantemente la **plataforma web** y sus funcionalidades. Al ser una marca nueva, puede llevar tiempo y esfuerzo establecerse en un mercado competitivo. Además, los **recursos financieros** iniciales necesarios para el desarrollo y marketing pueden ser **limitantes**.

A pesar de estas debilidades, hay muchas **oportunidades** para Game Nexus. La **industria** de los videojuegos está **en constante crecimiento**, lo que presenta oportunidades para captar nuevos clientes. La **creciente popularidad** de las comunidades de jugadores y las plataformas de social gaming es una tendencia que podemos aprovechar. Las **colaboraciones** estratégicas con influencers, desarrolladores de juegos y otras marcas

pueden ampliar nuestro alcance. Además, existe el potencial para expandir nuestra oferta a **mercados internacionales**, aprovechando la globalización de los videojuegos.

Enfrentamos algunas **amenazas**, como la **competencia intensa** de marcas establecidas como GameStop y EB Games, que ya tienen una base de clientes leal. La **rápida evolución tecnológica** significa que debemos adaptarnos rápidamente a los cambios en la industria de los videojuegos. También debemos **proteger la información** de los usuarios y mantener altos estándares de seguridad en la plataforma. Finalmente, los cambios en las **condiciones económicas y sociales** podrían afectar el gasto de los consumidores en entretenimiento.

En conclusión, Game Nexus tiene el potencial de convertirse en un líder en la venta de videojuegos y en un punto de encuentro central para la comunidad de jugadores. Al aprovechar nuestras fortalezas y oportunidades mientras gestionamos cuidadosamente nuestras debilidades y amenazas, podemos establecer una marca sólida y exitosa en la industria de los videojuegos.

Análisis DAFO de GameNexus	
Fortalezas	Debilidades
- Identidad de marca distintiva. - Enfoque comunitario. - Variedad de contenido. - Uso de tecnologías modernas.	- Dependencia de la tecnología para mantener y actualizar la plataforma. - Reconocimiento de marca inicial bajo en un mercado competitivo. - Recursos financieros iniciales limitados para desarrollo y marketing.
Oportunidades	Amenazas
- Crecimiento constante de la industria de los videojuegos. - Popularidad creciente de las comunidades de jugadores y social gaming. - Posibilidades de colaboración con influencers, desarrolladores de juegos y otras marcas. - Potencial para expansión internacional.	- Competencia intensa de marcas establecidas. - Rápida evolución tecnológica que requiere adaptación constante. - Riesgos de seguridad y privacidad de la información de los usuarios. - Cambios en las condiciones económicas y sociales.

1. Tabla de análisis DAFO.

5. Diseño web

5.1 Briefing

El proyecto tiene como objetivo desarrollar una **identidad de marca sólida y distintiva** para Game Nexus, una tienda de videojuegos, que refleje su pasión por la cultura del juego y su compromiso con la comunidad de jugadores.

La **personalidad** de la marca Game Nexus se define como **enérgica, innovadora, comunitaria, entusiasta e inmersiva**. Esta personalidad se debe reflejar en todos los aspectos de la comunicación y la experiencia del cliente.

Para lograr una **identidad visual** coherente y atractiva, se siguen las siguientes directrices de branding:

- **Colores:** Se seleccionará una paleta de colores vibrantes que refleje la emoción y la diversión de los juegos, con una preferencia por tonos llamativos pero equilibrados.
- **Logo:** Se diseñará un logo distintivo que combine elementos modernos y nostálgicos, transmitiendo la esencia de Game Nexus como un espacio de encuentro para la imaginación y la tecnología.
- **Tipografía:** Se utilizará una tipografía legible y moderna que se adapte a la personalidad de la marca, con una preferencia por formas audaces y contemporáneas.

El **tono de voz** de Game Nexus debe ser:

- **Conversacional:** Comunicarse de manera cercana y amigable con la audiencia, utilizando un tono accesible y familiar.
- **Entusiasta:** Transmitir la emoción y la pasión por los videojuegos en cada interacción, mostrando un compromiso genuino con la comunidad de jugadores.
- **Informativo:** Proporcionar información relevante y útil sobre productos, eventos y noticias de la industria de los videojuegos de manera clara y concisa.

La **experiencia de usuario** en Game Nexus, tanto en tienda como en línea, debe enfocarse en la **interactividad y accesibilidad**. Se diseñarán experiencias que fomenten la participación activa de los clientes y promuevan la **comunidad y el intercambio de ideas**. Además, se garantizará que la experiencia de compra sea fácil y accesible para todos los

clientes, independientemente de su nivel de experiencia en videojuegos o habilidades tecnológicas.

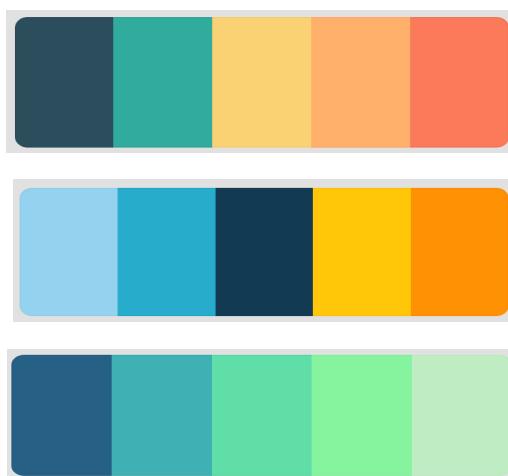
Para inspirar el **diseño y la identidad visual**, se tomarán como referencias marcas similares en la industria de los videojuegos, como GameStop, EB Games y tiendas especializadas en cultura geek, así como diseños inspiradores de logos e identidades visuales que capturen la **esencia de la marca** y su relación con la cultura del juego.

El éxito de la estrategia de branding se medirá mediante métricas como el incremento en la **participación de la comunidad** en eventos y actividades de Game Nexus, el **aumento en las ventas y la fidelidad de los clientes**, y el reconocimiento creciente de la marca y su asociación positiva con los valores y la cultura de los videojuegos.

5.2 Branding

El branding de Game Nexus se erige sobre la premisa de **la conexión y la inmersión** en un universo rico y diverso de experiencias digitales. A través de una combinación cuidadosa de elementos visuales, tonos de voz y experiencias de usuario, la marca busca no solo vender productos, sino también cultivar una comunidad vibrante y comprometida.

La **selección de colores** para la identidad visual de Game Nexus es un reflejo de su energía dinámica y su diversidad de contenido. Entre las paletas de colores consideradas, se destaca una que resuena especialmente con la esencia de la marca: una fusión de tonos vibrantes que evocan **la emoción y la aventura** de sumergirse en un mundo de juegos.



2. Selección de paleta de colores.

El logo de Game Nexus, la pieza central de su **identidad visual**, captura la esencia de la marca de manera distintiva y memorable. A través de un diseño que equilibra lo moderno con lo nostálgico, lo audaz con lo accesible, el logo transmite la idea de un espacio donde **la imaginación y la tecnología** se encuentran. Cada elemento del logo, desde su tipografía hasta sus íconos, está cuidadosamente elaborado para reflejar la pasión y el compromiso de Game Nexus con la excelencia en la experiencia de juego.



3. Selección logo de Game Nexus.

En resumen, el branding de Game Nexus no solo busca destacar en el competitivo mercado de las tiendas de videojuegos, sino también construir una identidad que resuene con la comunidad de jugadores, ofreciendo no solo productos, sino también un sentido de pertenencia y emoción compartida.

5.3 Mockups

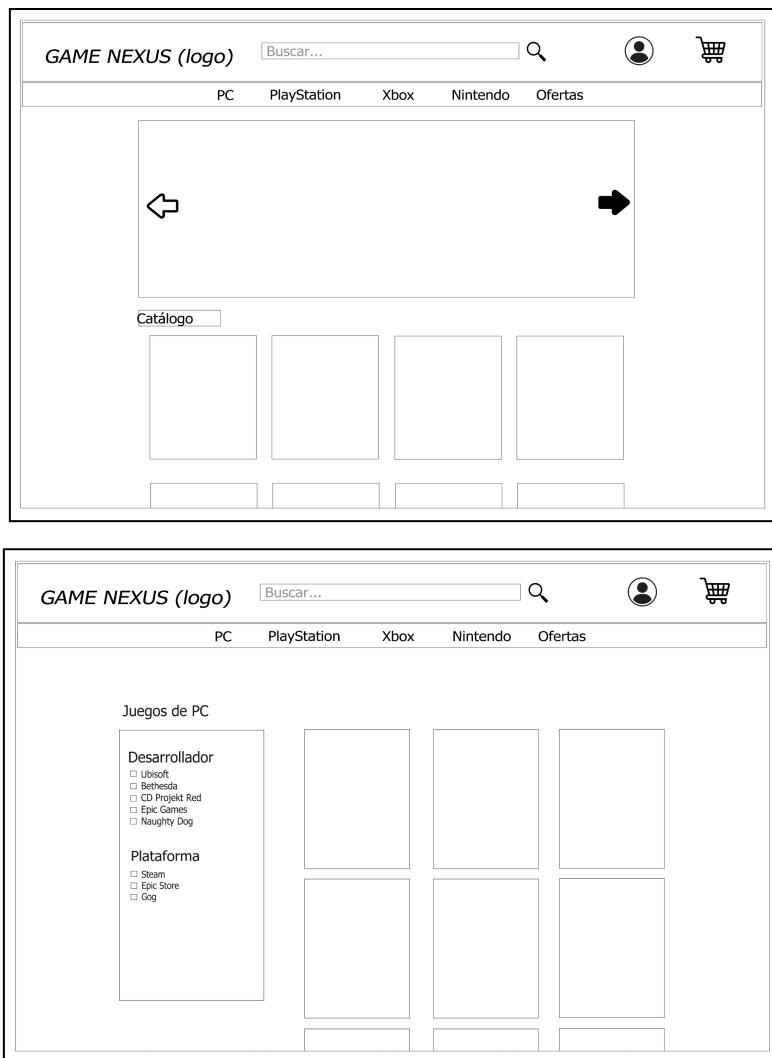
Los mockups son **representaciones gráficas** detalladas de la interfaz de la aplicación y ayudan a alinear la visión del equipo de desarrollo con las expectativas del cliente y los usuarios finales.

Para complementar el desarrollo de nuestra aplicación web Game Nexus, es fundamental diseñar y crear mockups que nos permitan visualizar **cómo será la interfaz de usuario y la experiencia del usuario (UX)** antes de implementar el código. A continuación, se describen algunos de los mockups esenciales para Game Nexus, una plataforma web dedicada a la comunidad de videojuegos.

Página de Inicio (Home)

La página de inicio es la primera impresión de la aplicación y debe capturar la atención de los usuarios. Los elementos clave incluyen:

- **Encabezado:** Con el logotipo de GameNexus y un menú de navegación que incluye un buscador y un filtro por plataformas para encontrar los juegos disponibles y enlaces para "Iniciar sesión" y "Comprar".
- **Banner Principal:** Una imagen o carrusel de imágenes destacadas de los juegos más populares o recientes, junto con breves descripciones y enlaces para más información.
- **Sección de Juegos Destacados:** Una galería de juegos recomendados con imágenes, títulos, y breves descripciones.
- **Noticias y Actualizaciones:** Un feed de las últimas noticias sobre videojuegos y actualizaciones importantes de la comunidad.
- **Pie de Página:** Información de contacto, enlaces a redes sociales, y otros recursos útiles.

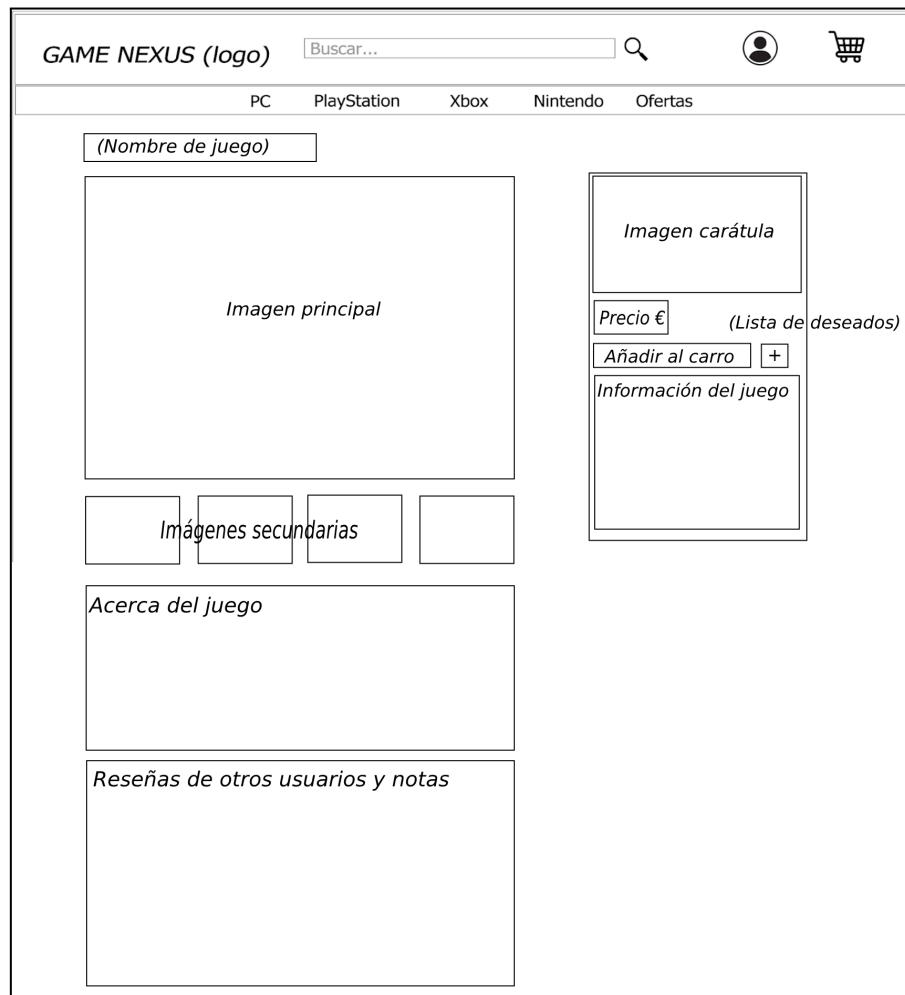


4. Mockups de la página principal.

Página de Detalle de Juego

Esta página proporciona información detallada sobre un juego específico. Los elementos clave incluyen:

- **Imagen y Título del Juego:** Una imagen grande del juego y su título prominente.
- **Descripción:** Una descripción completa del juego, incluyendo su género, desarrollador, y fecha de lanzamiento.
- **Características:** Lista de características destacadas del juego.
- **Valoraciones y Comentarios:** Sección donde los usuarios pueden dejar sus valoraciones y comentarios.
- **Botón de Acción:** Un botón para comprar o descargar el juego, o añadirlo a una lista de deseos.



5. Mockup de detalle de videojuego.

En resumen, los mockups de GameNexus son una parte esencial del proceso de desarrollo, proporcionando una **visualización clara del producto final y ayudando a identificar mejoras** antes de la implementación técnica. Con estos diseños, el equipo de desarrollo puede proceder con una **guía visual clara**, asegurando que todos los elementos de la aplicación estén alineados con las expectativas y necesidades del usuario.

6. Preparación del entorno de desarrollo

Para desarrollar una página web con Symfony, es fundamental instalar una serie de herramientas previas que facilitarán y optimizarán todo el proceso. A continuación, se presenta una **guía detallada sobre las herramientas necesarias y los pasos a seguir** para crear un proyecto Symfony desde cero.

En primer lugar, es necesario instalar **XAMPP**, un paquete de software que incluye Apache (servidor web), MySQL (sistema de gestión de bases de datos) y PHP (lenguaje de programación de servidores). XAMPP proporciona un **entorno de desarrollo local completo, esencial para ejecutar y probar aplicaciones PHP**, incluidas las creadas con Symfony. Se puede descargar XAMPP desde su sitio oficial y seguir el asistente de instalación para instalar Apache, MySQL y PHP.

A continuación, es imprescindible instalar **Composer**, un **administrador de dependencias para PHP** que facilita la instalación y gestión de las librerías y paquetes necesarios para el proyecto. Symfony utiliza Composer para gestionar sus dependencias y paquetes adicionales. Se puede descargar Composer desde su sitio oficial y seguir las instrucciones de instalación proporcionadas.

Otra herramienta crucial es **Symfony CLI**, una herramienta de línea de comandos que simplifica el desarrollo de proyectos Symfony. Esta herramienta **facilita la creación, configuración y administración de proyectos Symfony**. Symfony CLI se puede descargar desde su sitio oficial, donde también se pueden encontrar las instrucciones detalladas para su instalación.

Además, se debe instalar **Git**, un sistema de control de versiones que permite gestionar y coordinar cambios en el código fuente de un proyecto. Git es fundamental para **facilitar la colaboración en proyectos de software y el seguimiento de cambios** en el código. Para instalar Git, se descarga el instalador desde su sitio oficial y se siguen las instrucciones de instalación.

Asimismo, para simplificar aún más la instalación y gestión de estas herramientas, es recomendable utilizar **Scoop**, un gestor de paquetes para Windows. Scoop facilita la **instalación, actualización y administración** de programas y herramientas de desarrollo desde la línea de comandos.

Una vez instaladas todas las herramientas necesarias, se puede proceder a crear el proyecto Symfony desde cero. Es necesario abrir la línea de comandos (cmd) o PowerShell y ejecutar el siguiente comando para crear un nuevo proyecto Symfony: **symfony new nombre_del_proyecto --full**

Este comando creará una nueva estructura de proyecto con todas las dependencias y configuraciones iniciales necesarias. A continuación, se navega al directorio del proyecto y utilizamos Composer para instalar las dependencias del proyecto ejecutando el comando: **composer install**

El siguiente paso es configurar la base de datos. Abre **XAMPP** y asegurarnos de que **Apache** y **MySQL** estén ejecutándose. Luego, se crea una base de datos para el proyecto desde **phpMyAdmin** o utilizando la línea de comandos de MySQL. Abrimos el archivo `.env` en la raíz del proyecto Symfony y configuramos los detalles de la base de datos con la información concreta de nuestra base de datos. Una vez configurada la conexión, se ejecuta el siguiente comando para crear la base de datos: **symfony console doctrine:database:create**

Finalmente, para ejecutar el servidor web, se utiliza el siguiente comando: **symfony server:start**

En principio, esto iniciará el servidor web y la aplicación Symfony estará disponible en <http://localhost:8000> por defecto.

Siguiendo estos pasos, se configura un entorno de desarrollo completo para trabajar con Symfony en Windows. La utilización de herramientas como **XAMPP**, **Composer**, **Symfony CLI**, **Git** y **Scoop** no solo facilita el proceso de instalación y gestión de dependencias, sino que también asegura que el proyecto esté organizado y sea fácilmente mantible. La práctica y la exploración de la documentación oficial de cada herramienta te ayudarán a dominar cada una de ellas y a resolver cualquier problema que puedas encontrar durante el desarrollo de tu aplicación.

7. Desarrollo web

7.1 Estructura de la web

Para poder comprender bien la estructura de la web, tenemos que **separar las páginas en dos grupos**: públicas y privadas

Dentro de las **páginas públicas** tenemos:

- Index
- Vista única de producto
- Tienda
- Contacto y FAQ
- Sobre nosotros
- Mapa web
- Inicio de sesión
- Registro

Por otro lado, dentro de las **páginas privadas**, en las cuales se **requiere una autenticación del usuario** son:

- Panel de control
- Carrito

7.2 Funcionalidad

7.2.1 Index

Al entrar en la aplicación web la primera página que nos encontramos como usuario es “**Index**”, el cuál cuenta con una **galería de imágenes** desarrollada con JavaScript y la cual cuenta con dos modos de funcionamiento, mediante controles (flechas a cada lado) y arrastre. De esta forma el funcionamiento de la galería **se adapta perfectamente a todo tipo de dispositivos** haciendo más amigable la experiencia de usuario.

Por otro lado, tenemos **tres secciones claramente marcadas** donde se muestra un contenido dinámico extraído directamente de la base de datos. Estas tres secciones son: “*Los juegos más vendidos*”, “*Últimas ofertas*” y “*Últimas opiniones*”. En el caso de las dos primeras, **el usuario podrá agregar estos productos a su carrito y en todas ellas podrá acceder a los productos** seleccionados haciendo clic sobre las tarjetas.

Las tarjetas referentes a los productos y opiniones son generadas con un bucle en **Twig** que recorre los arrays entregados por la función “**index**” del controlador. A su vez **tenemos las imágenes en un array distinto** al anteriormente mencionado, por lo que **es necesario realizar una búsqueda comparando los id** para dar con la imagen correcta asociada a ese videojuego y plataforma. Este proceso se realiza en toda la web y gracias a que **con Symfony se trabaja con objetos** y en algunos casos incluso podemos acceder a otros objetos sin necesidad de realizar más de una petición a base de datos. En cambio en algunas ocasiones será necesario realizar queries más complejas que las que nos proporciona **Doctrine**, para ello se hace uso de los “**Service Entity Repository**” donde podremos crear funciones con queries más complejas o adaptarlas a nuestras necesidades. Las imágenes de las tarjetas de los videojuegos son obtenidas mediante el empleo de la función “**findOnePerGame()**” del repositorio “**MediaGamesRepository**”.

7.2.2 Registro y Login

Desde el index podemos acceder a dos de las páginas más importantes de la aplicación, **necesarias para poder acceder al resto de funcionalidades de la aplicación**, ya que necesitaremos el rol llamado “**ROLE_USER**”. Esta limitación **permite que usuarios no identificados accedan a áreas privadas** o se necesiten datos específicos de los mismos. Esto lo conseguimos **gracias al componente “Security” de Symfony**, lo que nos permite añadir rutas de control con el rol necesario para acceder a ellas. Los roles son asignados de manera automática a cada uno de los usuarios que se registre, junto a toda su información.

Tanto en el login como en el registro, y en **todas las vistas de formularios, se emplea el componente de Symfony** llamado “**FormBuilder**” para generar los formularios. Dentro de las clases de formularios, se asignan los inputs requeridos junto atributos como las clases, sus labels y el tipo de input. La **validación** de los campos se realiza **empleando el componente “Constraints” de Symfony**, realizando **comprobaciones comunes** como que los campos no sean nulos o estén vacíos, y en casos más específicos como el registro, que los datos sea únicos o coincidan con los filtros necesarios. Estos formularios son enviados al controlador “**SecurityController**”, donde se comprueba la validez de los datos. **En caso de que no sean válidos, son enviados de nuevo a la vista los errores visibles para el usuario.**

En el supuesto de que los **los datos del formulario de registro sean válidos**, se **inicia el proceso de generación del nuevo usuario junto con su wishlist personal y su carrito**. A su vez **ciframos la contraseña** del usuario empleando el **algoritmo Bcrypt** y guardamos todos los datos necesarios en la base de datos para luego redireccionar al usuario al index con la sesión ya iniciada.

Para el formulario de Login, emplearemos el componente **“LoginFormAuthenticator”** con el cual **Symfony valida automáticamente las credenciales** de nuestro usuario, **el token CSRF y la opción de recordar la cookie** de sesión (esta se recordará por un máximo de 15 días).

Si el proceso de autenticación es exitoso, se redirige al usuario al índice y en caso contrario se le muestran un mensaje de error en el formulario especificando cuál ha sido el problema empleando los mensajes **predefinidos de Symfony** o mensajes **personalizados** añadidos en el propio **FormBuilder**.

7.2.3 Tienda y productos

- Vista Individual de productos

Todos **los productos** de la tienda **tienen una vista independiente** donde el usuario podrá **ver información detallada** como imágenes, descripción, opiniones, pefi... etc. Cuando el usuario quiere acceder a un producto en concreto, **el controlador extrae y envía toda la información** referente al videojuego de la base de datos a una única plantilla de **Twig**. Para que el controlador conozca tanto el videojuego como la plataforma correspondiente, se extrae la información del slug de la ruta a la que el usuario desea acceder.

Los **slugs** de los videojuegos **están siempre compuestos por el nombre del videojuego y por su plataforma**. Por otro lado, **es necesario conocer si el usuario tiene este producto en su wishlist**. Por ello **se extrae el usuario de la sesión** y en caso de que no sea nulo, **se comprueba si el videojuego de la plataforma requerida está en su wishlist**. Una vez extraída toda la información necesaria, el controlador **envía los objetos necesarios con toda la información a la plantilla de Twig**.

La vista de la galería de imágenes de dentro los productos **está desarrollada en JavaScript con imágenes de distintos tamaños** y en formato Webp con el objetivo de

reducir el tiempo de carga. Esta galería **cuenta con dos ‘formatos’**, siendo el primero una galería con 4 pequeñas miniaturas en su parte inferior y dos flechas de control. En caso de tener más de 4 imágenes, desde JavaScript se actualizan según el usuario se va moviendo por ella. Por último, el segundo formato de galería se despliega en dispositivos con un viewport menor a 500 pixeles y este sería una galería donde el usuario puede ir pasando las imágenes arrastrándolas hacia la izquierda o derecha.

Para el sistema de reviews, dentro de la vista independiente de productos, **el usuario tiene la posibilidad de visualizar y generar reseñas del producto. Para que un usuario pueda crear una reseña, desde el controlador se verifica que este haya realizado mínimo una compra del producto** y que no tenga ya una reseña creada. Si esta condición se cumple, se manda a la vista el formulario de creación de reseñas, generado mediante “*FormBuilder*”.

Tanto en la sección de reseñas como en el resto de las secciones donde cabe la posibilidad de mostrar una gran cantidad de datos, **se emplea un paginador para distribuir estos resultados en distintas páginas**. Este paginador se encarga **desde el backend de añadir un limit y un offset a la query** inicial junto con tres métodos que nos devuelven los elementos de la página, el total de elementos de todas las páginas y la última página. **Desde JavaScript se controla la interacción del usuario con el controlador y se realizan las peticiones AJAX al servidor**, mostrando siempre un pequeño símbolo de carga durante el transcurso de la petición.

- Vista general de productos

El usuario puede navegar por los productos de la tienda accediendo a la vista general de productos. Esta sección **está dividida según el tipo de productos** que se muestran clasificando y facilitando la búsqueda de productos al usuario. De esta forma desde el menú, **el usuario podrá acceder a todos los videojuegos** de PC, PlayStation, Xbox, Nintendo, videojuegos con ofertas y a todos los videojuegos. Este proceso se realiza **haciendo una petición a base de datos empleando como filtro el slug de la vista**. Por otro lado, **los usuarios tienen la posibilidad de emplear filtros** con los que facilitar su navegación por la sección. **Cada filtro cuenta con su propio array en JavaScript, el cual se va llenando o vaciando según que el usuario marque o desmarque uno de los valores de cualquier filtro. Cuando se detecta un cambio en algún checkbox, se realiza una petición AJAX al controlador** enviando todos los arrays con su contenido.

Dentro del controlador se genera una query dinámica dependiendo del contenido de estos arrays y de esta forma obtenemos todas las características de los

videojuegos de la base de datos. El siguiente paso es realizar la búsqueda de las plataformas, filtro disponible en algunas de las vistas (PC, Todos y Ofertas).

Una vez hemos obtenido estos datos, podemos realizar la búsqueda en la base de datos, especificando todas las id de las características y de las plataformas, obteniendo así todos los productos que coincidan con estos parámetros especificados. Finalmente pasamos la query final por el paginador y realizamos una búsqueda de todas las imágenes necesarias empleando el id del videojuego y de la plataforma.

Por otro lado, el usuario también cuenta con la posibilidad de ordenar estos resultados en orden ascendente o descendente con respecto al precio o las valoraciones.

- Buscador

En la cabecera, el usuario tiene disponible una barra de búsqueda donde poder encontrar el videojuego que desea mediante su nombre. **Cada vez que el usuario realiza algún cambio en el contenido del input, desde JavaScript se realiza una petición AJAX a la función “searchBarResult()” del controlador “AjaxController”** enviando el contenido de la búsqueda. En caso de que ya haya una petición en ejecución, esta es cancelada automáticamente para evitar solapamientos de resultados. **Cuando la solicitud se complete, el usuario podrá ver pequeñas tarjetas** con los videojuegos que han coincidido con sus términos de búsqueda.

7.2.4 Sistema de carrito y compra

A cada usuario, como se ha mencionado anteriormente, se les genera un carrito propio automáticamente cuando estos se registran en la web. Así, desde el primer momento pueden comenzar a agregar productos a su carrito. Cuando el usuario hace click en alguno de los botones de agregar al carrito, **desde JavaScript se realiza una petición AJAX a la función “addProductCart()” del controlador “AjaxController”** enviando los datos del videojuego y plataforma. Primeramente, **se comprueba que el usuario esté correctamente autenticado**. En caso de que no sea así, se le redirige automáticamente al formulario de login. Si está autenticado, **comienza la búsqueda del videojuego en la base de datos y se comprueba que este tenga stock suficiente**, en caso de que no sea así se avisa al usuario mediante un mensaje de error.

Cuando se tiene un videojuego con suficiente stock, se realiza la búsqueda del carrito del usuario y **se llama a la función privada “*setProduct()*”**, donde se insertará el videojuego en caso de que este no esté ya en el carrito o se aumentará la cantidad si este se encuentra dentro. A su vez se modifica el total del carrito sumándole a este el precio con descuento incluido del producto.

Finalmente se aumenta la cantidad del carrito en la vista indicando que ahora tiene más productos dentro.

- Vista de carrito

Cada usuario tiene disponible una vista propia del carrito donde visualizar sus productos dentro de este, manipular sus cantidades y eliminarlos si este lo desea. Ambos procesos se realizan mediante **peticiones AJAX**.

Para la eliminación de un producto del carrito, se comprueba que el usuario esté autenticado y que tenga un carrito disponible. De ser así, se realiza la búsqueda del producto dentro del mismo para proceder a su eliminación. A su vez se procede a actualizar el total del carrito restándole el precio por la cantidad del producto eliminado y desde JavaScript actualizamos el front con los nuevos valores.

En el caso de la modificación de las cantidades, se realizan las mismas comprobaciones anteriormente mencionadas y una vez obtenemos los datos requeridos, se aumenta la cantidad. Si la cantidad es igual a 0, el producto es eliminado del carrito.

- Proceso de compra y asignación de keys

Para que el usuario pueda acceder al proceso de compra, este deberá tener productos en el carrito, una dirección de facturación y un método de pago.

Cuando el usuario cumple dichos requisitos y pulsa el botón de pagar, se llama a la función **“*payment()*”** del controlador del carrito. Antes de realizar el proceso, en el controlador se verifica que el método de pago y dirección de facturación pertenezcan al usuario, que estos no sean nulos y que el carrito no esté vacío. Si estas verificaciones se realizan de forma satisfactoria, comienza el proceso de comprobación de stock. Para ello se extrae de la base de datos todos los productos y se llama a la función privada **“*checkStock()*”**. Con esta función se **compara el stock disponible con la cantidad solicitada** por el usuario. **En caso de no haber suficiente stock, el controlador detiene el proceso de pago** y alerta al usuario de lo ocurrido, en cambio, si todos los productos

tienen stock suficiente, se genera un nuevo pedido con los datos del usuario, al que se le asocia el carrito con todos los productos y da comienzo al proceso de asignación de keys. En este proceso se recorre el array de productos y **se asignan tantas keys como cantidad de producto se haya adquirido**. Finalmente hacemos un **borrado lógico del carrito actual y generamos un nuevo carrito**.

7.2.5 Sistema de carrito y compra

Todos los usuarios registrados previamente tienen acceso a un panel de control personal, donde **puede gestionar toda su actividad dentro de la web**, ya sea modificando datos, visualizando pedidos, reviews, claves...

- Mis datos

Dentro de la sección de mis datos, **el usuario puede modificar su email, nombre de usuario y contraseña**. Para realizar esto se han implementado tres formularios distintos generados mediante “*FormBuilder*”. Tanto el input de email como del nombre de usuario tienen por defecto el valor actual de estos apartados. En ambos casos, siempre que el usuario realiza la modificación, **se comprueba que no están en uso por otros usuarios para evitar duplicidad** de campos únicos y claves primarias en la base de datos **junto con la comprobación del token CSRF**. Cuando las validaciones son correctas, se realiza la modificación del parámetro y en el caso del email, antes de redirigir la petición nuevamente a la vista de mis datos, **se debe realizar la autenticación del usuario desde el controlador** ya que *Symfony* identifica a cada usuario mediante su email de forma única(ya que es clave primaria).

En el caso de la contraseña, seguimos un proceso parecido al de los otros formularios, pero aquí contamos con un input para la contraseña actual y dos campos de repetición para la nueva contraseña, los cuales, obviamente deben coincidir. Desde el controlador se verifica que la contraseña actual sea válida y **se procede a cifrar la nueva contraseña empleando el algoritmo *Bcrypt*** para después actualizar el usuario en la base de datos. Si durante el proceso la página detecta que hay algún fallo en alguna de las peticiones se muestra al usuario un mensaje de error.

Finalmente regeneramos la cookie de sesión para poder usar nuestro nuevo correo y nombre de usuario, mostrando estos en la vista.

- Métodos de pago y dirección de facturación

Para realizar acción de eliminar, tanto con el método de pago y la dirección de facturación, el usuario debe pulsar sobre la cruz de cualquiera de las tarjetas para que desde JavaScript se acceda al atributo que contiene el id y una vez obtenido se realiza una petición AJAX a las funciones “**deleteCard()**” (En caso de la tarjeta de crédito) o “**deleteBilling()**” (En el caso de las direcciones de facturación). Con dichas funciones se realiza la búsqueda en base de datos del objeto empleando tanto su id como el id de usuario. De esta forma podemos verificar que el método de pago o dirección de facturación pertenecen al usuario autenticado y se procede a realizar un borrado de todos los campos, desapareciendo también en la vista gracias a JavaScript una vez la petición ha sido completada.

- Formularios de pago y dirección de facturación

Como se ha comentado anteriormente, todos los formularios están generados con “*FormBuilder*”. Además de lo que nos proporciona el “*FormBuilder*” añadimos más validaciones para los métodos de pago, donde se verifica que las tarjetas tengan el formato de VISA, MASTERCARD y MAESTRO empleando el constraint “**Cardscheme**”. Por otro lado, la fecha de validez es verificada empleando una expresión regular adicional con el constraints “**Regex**”.

7.2.6 Panel de control

- Wishlist

Desde la sección de wishlist, el usuario puede visualizar, eliminar y agregar al carrito los juegos que están dentro de la misma. Los videojuegos se muestran de 6 en 6 empleando un paginador. **Para la eliminación de un producto de la wishlist, se realiza una petición AJAX a la función “**deleteWishlistGame()**”**, desde donde verificamos que el usuario esté autenticado y el producto pertenezca a su wishlist para acto seguido eliminarlo de la tabla en la base de datos. Finalmente se redirige la petición a la función “**wishlistGames()**” del mismo controlador para actualizar la vista en tiempo real.

Desde la vista de wishlist, el usuario también tiene la posibilidad de agregar cualquier producto de su wishlist al carrito siguiendo el mismo proceso que hemos comentado con anterioridad en otras páginas.

- Reseñas

Todas las reseñas generadas por el usuario pueden ser visualizadas desde la vista reseñas de su panel de control. Mediante el empleo de un paginador, se muestran de 6 en 6 (al igual que con los productos de la wishlist) junto con la imagen y título del videojuego, plataforma, contenido de la reseña y calificación de esta.

- Pedidos

Los usuarios pueden acceder a la información de sus pedidos desde la vista de pedidos. Mediante el empleo de un paginador se mostrarán de 6 en 6 los pedidos junto con los videojuegos comprados.

Las tarjetas se muestran en formato de acordeón, permitiendo una visualización cómoda de los pedidos y de todo su contenido evitando saturar la vista.

- Visualización de keys

Las claves de cada pedido pueden ser visualizadas pulsando al botón ‘ver claves’ de cada producto dentro de pedidos, enviando al usuario a la vista de keys donde aparecerá la fecha de compra, el nombre e imagen del videojuego, cantidad de keys y un botón que permite copiar la key al portapapeles mediante JavaScript.

- Cerrar sesión

El usuario puede cerrar la sesión desde el botón ‘cerrar sesión’ ubicado en el menú del panel de control. Este proceso es realizado **automáticamente por Symfony si se le indica en el archivo “security.yaml”** la ruta, o el nombre de la ruta, de la función de “*logout()*”, la cual no es necesario llenar.

7.3 Casos de uso

El usuario ingresa en la página web Game Nexus y lo primero que visualiza es la **interfaz principal** que consta de varios elementos, destacando su amplia **selección de videojuegos** y **funcionalidades de búsqueda avanzadas**. La estructura y los elementos clave de la página principal incluyen en el encabezado el logo de la empresa, una **barra de búsqueda por nombre** del videojuego, un botón para iniciar sesión y un botón para ver el carrito.

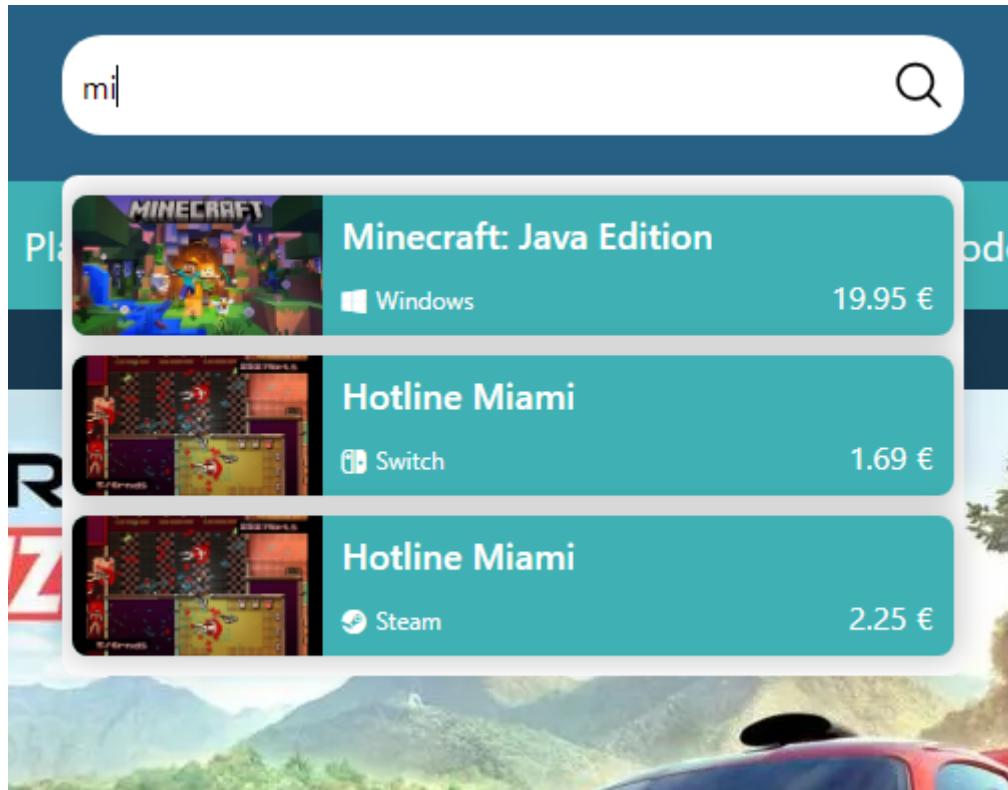
Además, hay un **navegador** que permite **clasificar los videojuegos por plataforma** para facilitar al usuario una búsqueda más exhaustiva de lo que está buscando y más abajo, un **banner de anuncios y promociones** que el usuario puede ver utilizando las flechas disponibles o arrastrando el banner como se haría en un móvil.

The screenshot shows the homepage of Game Nexus. At the top, there is a navigation bar with a search bar, user account links ('Iniciar sesión'), and a shopping cart icon ('Ver carrito'). Below the header, there are links for PC, PlayStation, XBOX, Nintendo, Ofertas (Offers), and Todos los productos (All products). A large banner for the game 'FORZA HORIZON 5' is prominently displayed, featuring a red sports car driving through a lush, tropical landscape. Below the banner, a section titled 'Los más vendidos' (Top sellers) displays a grid of eight game cards. Each card includes the game title, price, a rating (e.g., 6.5, 20.39, 0.0), and a 'Añadir al carrito' (Add to cart) button. The games listed are Minecraft: Java Edition, Bioshock: The Collection, Assetto Corsa, Sniper Elite 5, Sniper Elite 5, Call of Duty Modern Warfare, Hotline Miami, and Grand Theft Auto V. At the bottom of the page, there is a horizontal row of smaller game thumbnails.

Game	Platform	Price	Rating	Action
Minecraft: Java Edition	Windows	19.95€	6.5	Añadir al carrito
Bioshock: The Collection	Switch	20.39€	20.39€	Añadir al carrito
Assetto Corsa	Steam	19.99€	0.0	Añadir al carrito
Sniper Elite 5	PlayStation	60.00€	7.3	Añadir al carrito
Sniper Elite 5	Steam	50.00€	0.0	Añadir al carrito
Call of Duty Modern Warfare	Steam	29.99€	0.0	Añadir al carrito
Hotline Miami	Steam	2.25€	0.0	Añadir al carrito
Grand Theft Auto V	Steam	18.05€	0.0	Añadir al carrito

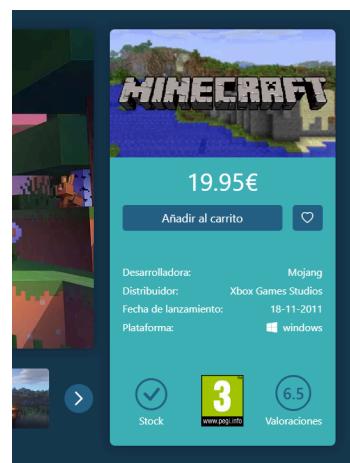
6. Interfaz del index o página principal.

La **búsqueda por nombre** despliega una lista del juego en cuestión directamente en pequeñas tarjetas. Si se selecciona una de ellas, se muestra una página del juego donde se puede visualizar una **descripción del mismo, los requisitos necesarios y opiniones** en caso de que algún usuario que haya comprado el juego haya hecho una reseña.



7. Motor de búsqueda por nombre.

En este apartado donde se visualiza el juego, se puede **añadir a la lista de deseos del usuario** y, en caso de querer realizar la compra, **añadir al carrito**.



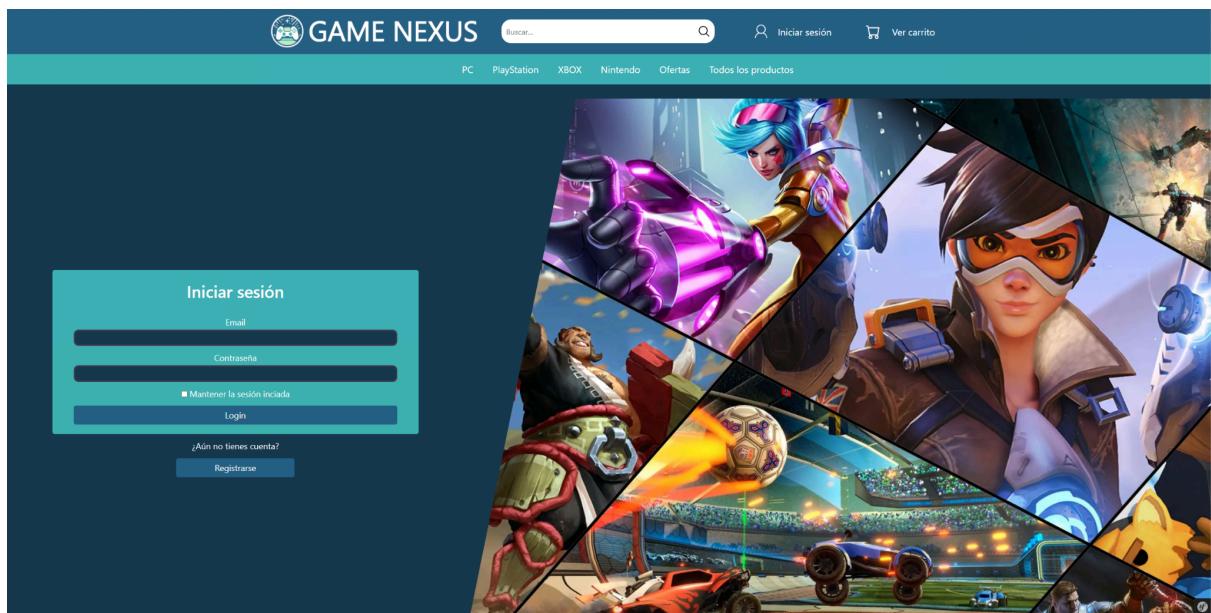
8. Página de detalle del juego, específicamente, su tarjeta funcional.

Como se puede observar, el usuario puede **recorrer la página web libremente** sin haber iniciado sesión, sin embargo, en caso de querer realizar alguna de las anteriores acciones, **comprar o añadir juego a la lista de deseos, debe iniciar sesión o en su defecto, registrarse**, . De hecho, se le redirige de forma automática para que comience ese proceso.

The screenshot shows a teal-colored registration form titled "Registro". It contains five input fields: "Email", "Nombre de usuario", "Contraseña", "Repetir contraseña", and a blue "Registrarse" button at the bottom.

9. Detalle de página de registro.

A continuación, el usuario se **registroaría** metiendo los datos necesarios y posteriormente, en caso de estar correctos, **iniciaría sesión** introduciendo sus credenciales.



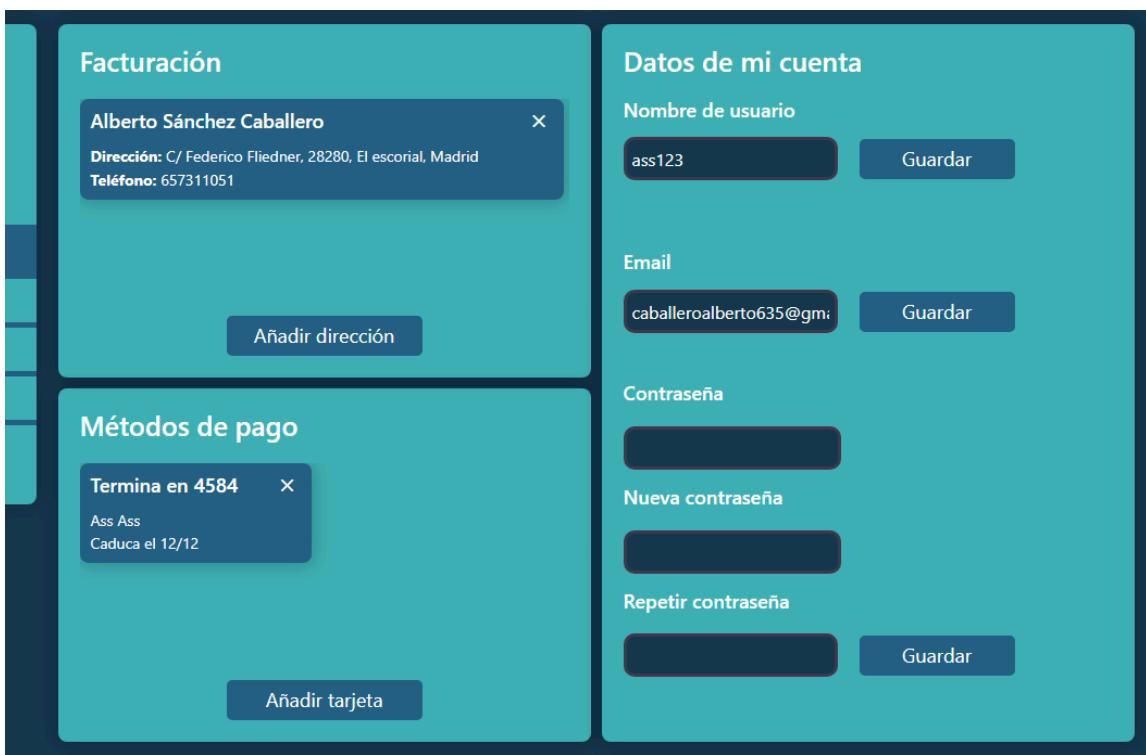
10. Página de iniciar sesión.

Cuando el usuario **inicia sesión**, en la página principal se sustituye el botón de “iniciar sesión” por el perfil con una fotografía por defecto. Desde ahí, el usuario puede **acceder a su perfil y realizar distintas acciones**. Como es de esperar, en el perfil el usuario podrá **cerrar la sesión** si lo desea.



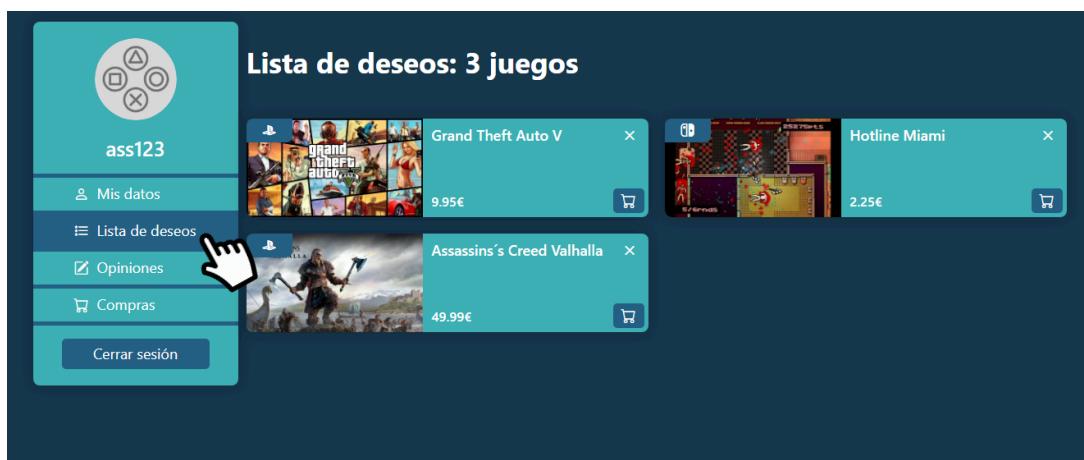
11. Detalle de acciones del perfil.

En primer lugar, el usuario tiene un apartado de “mis datos” donde debe añadir una **dirección** y un **método de pago** como mínimo para poder realizar compras. Además, tiene un apartado con los **datos de la cuenta**, nombre, email y contraseña. Cualquiera de ellos puede ser modificado por el usuario.



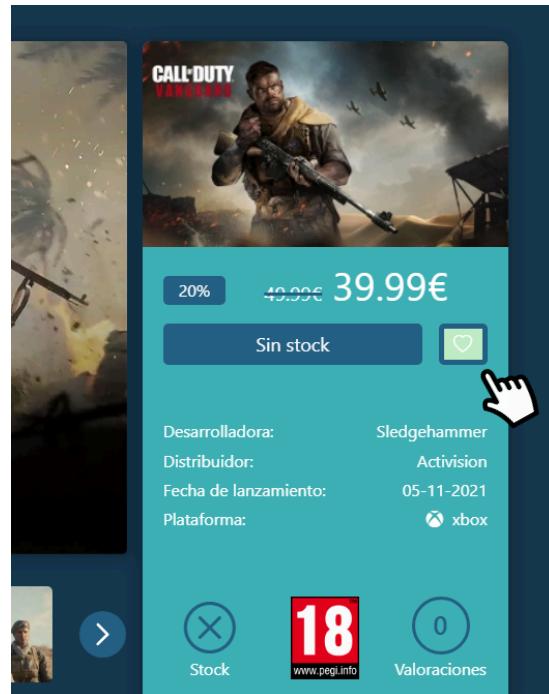
12. Pestaña de “Mis datos”.

Por otro lado, Game Nexus ofrece a sus usuarios la opción de crear **una lista de deseos**, permitiéndoles guardar y gestionar sus juegos favoritos de manera sencilla y organizada. Esta funcionalidad no solo **facilita el seguimiento** de títulos de interés, sino que también permite a los usuarios **planificar futuras compras y recibir notificaciones sobre descuentos y disponibilidad** de sus juegos deseados.



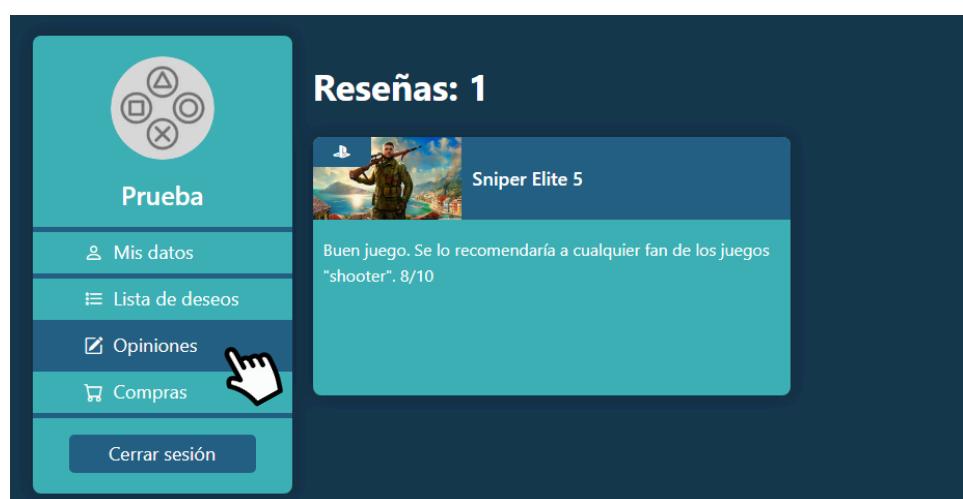
13. Pestaña de “Lista de deseos”.

Para poder añadir un juego a la lista, el usuario tiene que meterse en el juego que quiere y desde ahí pinchar en el **botón de “me gusta”** con un símbolo de corazón. La próxima vez que se meta a la lista en su perfil **podrá ver los juegos añadidos** en la lista de deseos y borrarlos pinchando en la cruz, en caso de no estar más interesado.



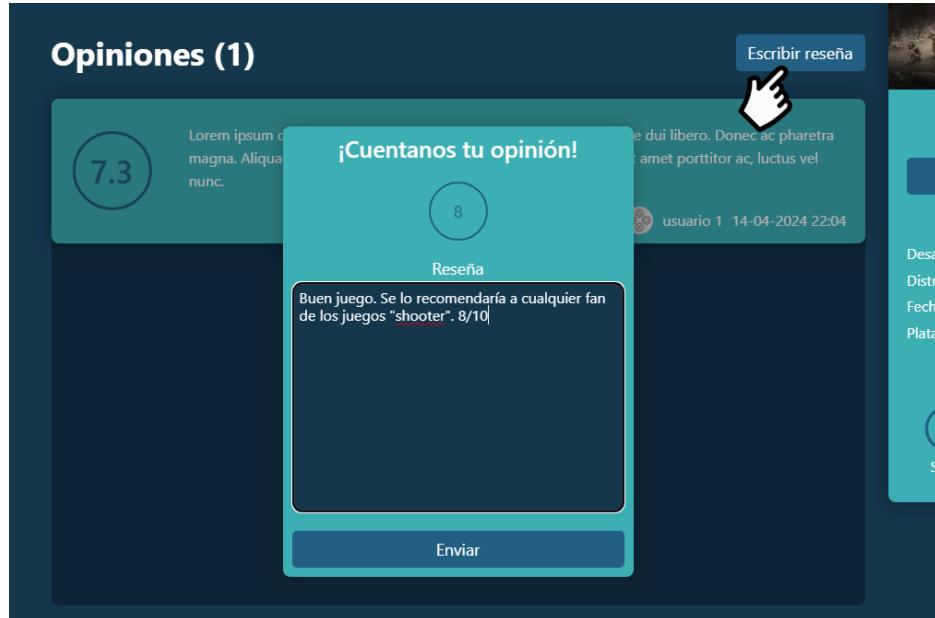
14. Acción “añadir a la lista de deseos” en el detalle del videojuego.

En el perfil, el usuario también podrá **ver las reseñas u opiniones** que haya realizado. Esta opción de realizar una reseña sólo será factible **en caso de que el usuario haya hecho la compra** en la plataforma y que, por tanto, haya jugado.



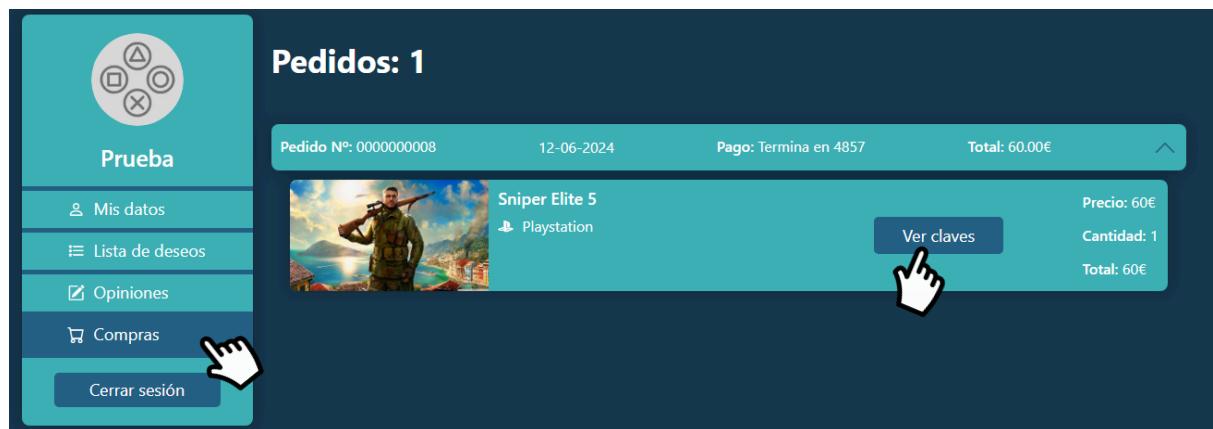
15. Pestaña de “Opiniones”.

Para realizar una reseña el usuario debe meterse en el juego comprado y pinchar en un nuevo botón que aparece “**Escribir reseña**”. Ahí el usuario podrá escribir **su opinión y darle una puntuación**. Sólo se puede hacer una reseña por compra. De esta manera, las puntuaciones dependen única y exclusivamente de la comunidad.



16. Acción “Escribir reseña” en el detalle del videojuego.

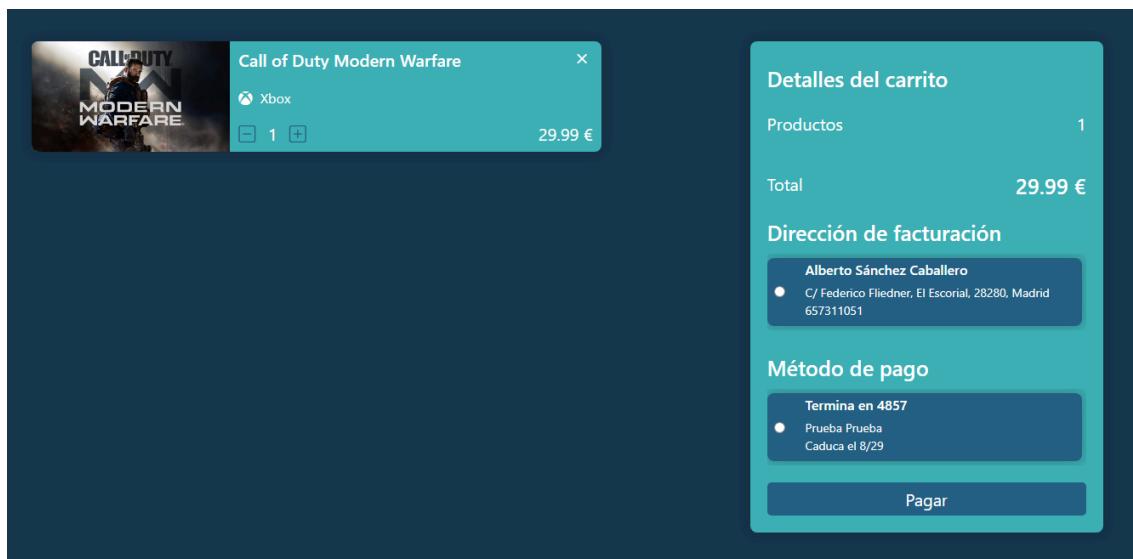
Por último, el usuario desde el perfil podrá ver sus **compras o pedidos realizados**, con las **claves** que deberá utilizar para poder comenzar a jugar. En caso de tener dudas sobre cómo proceder, se ha añadido un **enlace** que redirige al usuario a una página donde se contestan algunas preguntas frecuentes.



17. Acción “Ver claves” en la pestaña “Compras”.

Evidentemente, al tratarse de una tienda online, el usuario que haya iniciado sesión podrá **realizar compras**. Para ello, debe navegar por la página web e ir añadiendo al carrito los juegos que deseé comprar. **Cada uno de ellos se irá añadiendo al carrito, al cual podremos acceder desde cualquier punto de la página web.**

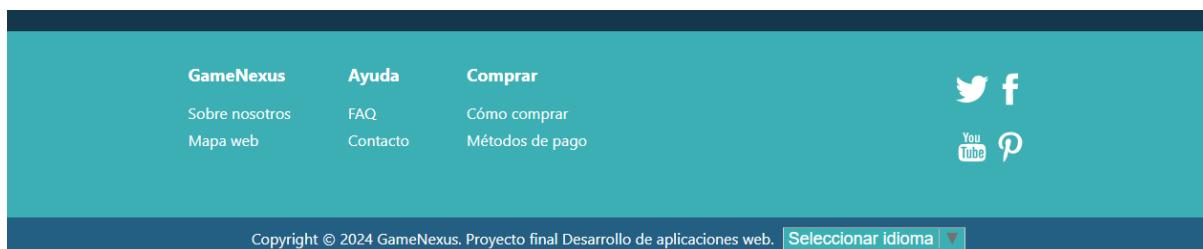
Al acceder al carrito, el usuario podrá **ver sus juegos añadidos y quitarlos** si ha cambiado de opinión, cabe destacar que si no hay stock no podrá añadirlo. En los detalles del carrito, aparecen el **número de productos añadidos, el precio total y las direcciones de facturación así como los métodos de pago** que haya incluido en su perfil, los cuales puede seleccionar según sus preferencias y proceder al pago pulsando el botón “Pagar”.



18. Página “Ver carrito” y acción “Pagar”.

Cuando pulse el botón se le redirige a la página principal, podrá ver su pedido en el perfil en el **apartado de compras**, donde podrá recoger la clave de forma inmediata que ya estaría disponible para **comenzar a disfrutar** de su juego.

Como detalles extra, el **pie de página** incluye varios enlaces que llevan al usuario a páginas por si quiere **conocer más sobre la empresa o tiene algunas dudas** de la misma. Además, incorpora un desplegable por si el usuario quisiera **cambiar el idioma** de la página. Se trata de un plugin de google que se ha configurado para mostrar 7 idiomas más aparte del original.



19. Pie de página y selección de idioma.

7.4 Portabilidad

La **portabilidad** y la **adaptabilidad** de Game Nexus son aspectos cruciales para asegurar que la plataforma brinde una **experiencia de usuario** óptima en una **variedad de dispositivos**. Esta decisión de hacer que Game Nexus sea una página web “responsive” no solo ofrece numerosas ventajas para los usuarios sino también para la propia empresa.

Al ser una página web responsive, Game Nexus permite que los usuarios accedan a la plataforma desde cualquier dispositivo, ya sea una computadora de escritorio, laptop, tableta o teléfono móvil. Esto es crucial para alcanzar una **audiencia amplia y diversa**, que puede preferir diferentes dispositivos para interactuar con la plataforma. La **accesibilidad mejorada** garantiza que todos los usuarios, independientemente del dispositivo que utilicen, puedan disfrutar de una experiencia de usuario consistente y satisfactoria.

Además, asegura que el diseño y la funcionalidad se adaptan automáticamente al tamaño de la pantalla del dispositivo utilizado. Esto significa que las imágenes, textos y botones se redimensionan y reubican de manera adecuada, **evitando problemas** de usabilidad como el desplazamiento horizontal o el zoom constante. Esta **consistencia** en la experiencia de usuario es fundamental para mantener a los usuarios comprometidos y satisfechos, sin importar desde qué dispositivo accedan.

Las páginas web responsive están optimizadas para cargar rápidamente en todos los dispositivos, lo que **mejora significativamente el rendimiento y reduce las tasas de rebote**. Los tiempos de carga rápidos son esenciales para mantener la atención de los usuarios y prevenir que abandonen el sitio debido a largos tiempos de espera. Una navegación ágil y eficiente contribuye a una experiencia de usuario positiva y aumenta la probabilidad de que los visitantes permanezcan en Game Nexus y realicen compras.

Por otro lado, los motores de búsqueda, como Google, prefieren sitios web que sean compatibles con dispositivos móviles. Al ser responsive, Game Nexus **mejora su posicionamiento** en los resultados de búsqueda, lo que **aumenta su visibilidad y atrae más tráfico** a la plataforma. Una mayor visibilidad en los motores de búsqueda se traduce

en un mayor alcance, lo que puede resultar muy útil para aumentar las ventas de nuestra empresa.

Otra de las ventajas de Game Nexus es su versión única de página web, ya que mantener una única versión de la web que se adapta a todos los dispositivos es **más eficiente y menos costoso** que desarrollar y mantener versiones separadas para escritorio y móviles. Con una página web responsive, cualquier actualización o cambio se aplica a todas las versiones de la web simultáneamente, lo que simplifica el proceso de mantenimiento y reduce los costos operativos.

Por último, el diseño responsive no solo se adapta a los dispositivos actuales, sino que también está **preparado para futuros dispositivos** con diferentes tamaños y resoluciones de pantalla. Esta adaptabilidad asegura que Game Nexus se **mantenga relevante y accesible**, independientemente de cómo evolucionen las tecnologías y los dispositivos en el.

7.5 Rendimiento

En el desarrollo de Game Nexus, los **test unitarios** juegan un papel crucial para asegurar que los **casos de uso se implementen correctamente y funcionen** según lo previsto. Los test unitarios son una técnica de prueba de software que verifica la funcionalidad de componentes individuales del código, como funciones o métodos, de manera aislada.

El **objetivo principal** de los test unitarios en Game Nexus es **validar** que cada parte del **sistema opere correctamente y de manera independiente**. Esto incluye verificar la lógica de negocio, las interacciones con la base de datos y las respuestas a las peticiones del usuario. Los test unitarios **permiten detectar errores** en las primeras etapas del desarrollo, lo que facilita su corrección antes de que se integren en el sistema completo.

Durante el proceso de desarrollo de Game Nexus, se ha realizado una **tabla descriptiva general** de distintos test unitarios con el objetivo de **asegurar la funcionalidad y la integridad del sistema**. Esta tabla sirve como una guía estructurada que detalla los casos de uso específicos que cada test unitario debe verificar, proporcionando una **visión clara y organizada del proceso** de pruebas.

Nº Caso de Prueba	Descripción	Resultados Esperados	Obtenidos	Finales
1	Iniciar sesión	Usuario debe ser redirigido a la página de inicio después de ingresar credenciales válidas.	Usuario fue redirigido a la página de inicio después de ingresar credenciales válidas.	OK
2	Registrarse	Usuario debe ser redirigido a su carrito.	Usuario fue redirigido a su carrito después de registrarse.	OK
3	Buscar por categoría	Deben mostrarse productos correspondientes a la categoría seleccionada.	Se mostraron productos correspondientes a la categoría seleccionada.	OK
4	Buscar en el buscador	Deben mostrarse resultados relevantes a la palabra clave ingresada en el buscador.	Se mostraron resultados relevantes a la palabra clave ingresada en el buscador.	OK
5	Añadir un producto al carrito	El producto debe ser añadido al carrito y el usuario debe ver que el producto ha sido añadido.	El producto fue añadido al carrito y el usuario vio la confirmación de adición.	OK
6	Quitar un producto del carrito	El producto debe ser eliminado del carrito y el usuario debe ver que el producto ha sido eliminado.	El producto fue eliminado del carrito y el usuario vio la confirmación de eliminación.	OK
7	Ver perfil de usuario	El usuario debe poder acceder a su perfil, añadir datos y métodos de pago.	El usuario pudo acceder a su perfil, añadir datos y métodos de pago.	OK
8	Ver lista de deseos	El usuario debe ver la lista de los juegos favoritos a los que ha dado "me gusta".	El usuario vio la lista de los juegos favoritos a los que dio "me gusta".	OK
9	Ver opiniones realizadas	El usuario debe ver las opiniones creadas.	El usuario vio las opiniones que creó.	OK
10	Pagar	El sistema debe aceptar la tarjeta válida y mostrar un mensaje de confirmación.	El sistema aceptó la tarjeta válida y mostró un mensaje de confirmación.	OK
11	Comprar	La compra debe ser procesada y el usuario debe recibir una confirmación de la transacción.	La compra fue procesada y el usuario recibió una confirmación de la transacción.	OK
12	Pedidos	El usuario debe ver los pedidos realizados en su perfil.	El usuario vio los pedidos realizados en su perfil.	OK
13	Multiidioma	La página web debe cambiar al idioma seleccionado y mostrar todo el contenido en ese idioma.	La página web cambió al idioma seleccionado y mostró todo el contenido en ese idioma.	OK

20. Resumen de test unitarios.

Los test unitarios son fundamentales para asegurar que Game Nexus **opere de manera fiable y eficiente**. Al implementar una estrategia robusta de pruebas unitarias, se mejora la calidad del software, se reduce el tiempo de desarrollo y se garantiza una experiencia de usuario consistente y satisfactoria.

8. Diagrama de Gantt

El **primer bloque del diagrama** sobre el proyecto de desarrollo de game nexus incluye la generación de ideas, diseño de mockups, preparación del entorno de desarrollo, configuración del repositorio de GitHub, creación de la base de datos y desarrollo inicial de la página principal. Esta etapa **establece las bases técnicas y visuales** del proyecto.

21. Primer bloque: bases técnicas y visuales.

El **segundo bloque** del proyecto incluye el estudio de la documentación de Symfony, la creación de la base de datos, el desarrollo de la estructura y lógica de la página principal, la aplicación de estilos iniciales con SCSS y la documentación del proyecto. Esta etapa se **enfoca en el desarrollo y documentación técnica** del proyecto.

22. Segundo bloque: desarrollo y documentación técnica.

El **tercer y último bloque** del proyecto incluye el desarrollo de las páginas de menú y juegos, la gestión de la base de datos, la estilización de las páginas de menú y juegos con SCSS, la implementación del inicio de sesión y gestión de usuarios, así como la gestión del carrito y la lista de deseados. Esta etapa se **enfoca en la funcionalidad avanzada y la experiencia del usuario**.

23. Tercer bloque: funcionalidad avanzada y experiencia de usuario.

9. Conclusión del proyecto

En conclusión, la página web de Game Nexus se destaca por su diseño intuitivo y funcionalidades pensadas para mejorar la experiencia del usuario. Con una barra de búsqueda avanzada, los usuarios pueden encontrar fácilmente los videojuegos que desean por título, género, plataforma y otros filtros. Además, la posibilidad de crear una lista de deseos permite a los usuarios guardar y gestionar sus juegos favoritos, facilitando el seguimiento de títulos de interés y la planificación de futuras compras.

La estructura de la página principal, que incluye categorías destacadas, imágenes de alta calidad y secciones dedicadas a promociones y lanzamientos recientes, asegura una navegación fluida y atractiva. En conjunto, estos elementos hacen de Game Nexus un sitio web completo y accesible, ideal para los entusiastas de los videojuegos que buscan una plataforma eficiente y bien organizada.

10. Bibliografía

1. Symfony. (s.f.). *The Symfony documentation.* Recuperado de <https://symfony.com/doc/current/index.html>
2. Symfony. (s.f.). *Symfony 6.0 documentation.* Recuperado de <https://symfony.com/doc/6.0/index.html>
3. Symfony. (s.f.). *Setting up Symfony with Flex.* Recuperado de <https://symfony.com/doc/6.0/setup/flex.html>
4. Symfony. (s.f.). *Page creation.* Recuperado de https://symfony.com/doc/6.0/page_creation.html
5. Symfony. (s.f.). *Debug mode.* Recuperado de https://symfony.com/doc/6.0/configuration/front_controllers_and_kernel.html#debug-mode
6. Symfony. (s.f.). *HTTP cache.* Recuperado de https://symfony.com/doc/6.0/http_cache.html#symfony-gateway-cache
7. Symfony. (s.f.). *Web server configuration.* Recuperado de https://symfony.com/doc/6.0/setup/web_server_configuration.html
8. MDN Web Docs. (s.f.). *Responsive design.* Recuperado de https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Responsive_Design
9. MDN Web Docs. (s.f.). *The web and web standards.* Recuperado de https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/The_web_and_web_standards
10. Doctrine Project. (s.f.). *Doctrine ORM 2.11 documentation.* Recuperado de <https://www.doctrine-project.org/projects/doctrineorm/en/2.11/index.html>
11. Eneba.(s.f.). *Imágenes de videojuegos.* Recuperado de <https://www.eneba.com/es/>
12. Bootstrap. (s.f.). *Iconos.* Recuperado de <https://icons.getbootstrap.com/>
13. Roams. (s.f.). *Operadora - sobre nosotros.* Recuperado de <https://roams.es/companias-telefonicas/pepephone/atencion-cliente/>
14. Epic Games. (2023). *The Epic Games Store Summer Sale kicks off with savings up to 75% off select titles.* Recuperado de <https://store.epicgames.com/en-US/news/the-epic-games-store-summer-sale-kicks-off-with-savings-up-to-75-off-select-titles>
15. Goodyear. (s.f.). *Coches Need for Speed.* Recuperado de <https://kilometrosquecuentan.goodyear.eu/coches-need-for-speed/>

16. Slickdeals. (s.f.). *Horizon Forbidden West best deals*. Recuperado de <https://daily.slickdeals.net/tech/horizon-forbidden-west-best-deals/>
17. Liukin. (s.f.). *Forza Horizon 5: Todo lo que sabemos*. Recuperado de <https://liukin.es/forza-horizon-5-todo-lo-que-sabemos/>
18. G2A (s.f.). Descripciones de videojuegos. Recuperado de <https://www.g2a.com/es/>